Computing Persistent Homology with Various Coefficient Fields in a Single Pass

Jean-Daniel Boissonnat, Clément Maria

Computing Persistent Homology with Various Coefficient Fields in a Single Pass *

Jean-Daniel Boissonnat and Clément Maria

INRIA Sophia Antipolis-Méditerranée, France
{jean-daniel.boissonnat, clement.maria}@inria.fr

Abstract. This article introduces an algorithm to compute the persistent homology of a filtered complex with various coefficient fields in a single matrix reduction. The algorithm is output-sensitive in the total number of *distinct* persistent homological features in the diagrams for the different coefficient fields. This computation allows us to infer the prime divisors of the torsion coefficients of the integral homology groups of the topological space at any scale, hence furnishing a more informative description of topology than persistence in a single coefficient field. We provide theoretical complexity analysis as well as detailed experimental results. The code is part of the `Gudhi` library, and is available at [8].

1 Introduction

Persistent homology [5,12] is an algebraic method for measuring the topological features of the sublevel sets of a function defined on a topological space. Its generality and stability [4] with regard to noise have made it a widely used tool for the study of data. At the algebraic level [12], it admits a decomposition – represented by mean of a persistence diagram – only when considered with field coefficients (by opposition to integer coefficients). The persistence diagram contains a rich information about the topology of the studied space and very efficient methods exist to compute it. However, the integral homology groups of a topological space are strictly more informative than the homology groups with field coefficients, in particular because they convey information about "torsion". Torsion can be pictured geometrically as a "twisting" of the shape and happens frequently as global topological feature in topological data analysis where, for example, Klein bottles appear naturally [3,9]. Algebraically, torsion is characterized by cyclic subgroups of the integral homology groups. When computed with field coefficients, these subgroups may either vanish or appear as "infinite", and consequently obfuscate the study of the topology of data. A simple solution is to compute persistent homology with different coefficient fields and track the differences in the persistence diagrams.

* This research has been partially supported by the European Research Council under Advanced Grant 339025 GUDHI (Algorithmic Foundations of Geometric Understanding in Higher Dimensions).

We build on this idea and describe an algorithm to compute persistent homology with various coefficient fields $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ in a single pass of the matrix reduction algorithm, where $\mathbb{Z}_q$ denotes the finite field $\mathbb{Z}/q\mathbb{Z}$ for a prime q. To do so, we introduce a method we call *modular reconstruction* consisting in using the *Chinese Remainder Isomorphism* to encode an element of $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ with an element of $\mathbb{Z}_{q_1 \cdots q_r}$. We describe algorithms to perform elementary row/column operations in a matrix with $\mathbb{Z}_{q_1 \cdots q_r}$ coefficients, corresponding to simultaneous elementary row/column operations in matrices with coefficients in the fields $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$. The method results in an algorithm with an output-sensitive complexity in the total number of *distinct* pairs in the echelon forms of the matrices with $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ coefficients, plus an overhead due to arithmetic operations on big numbers in $\mathbb{Z}_{q_1 \cdots q_r}$. The method is generic and applies to every algorithm for persistent homology. Finally, we describe how to infer the torsion coefficients of the integral homology using the *Universal Coefficient Theorem for Homology*.

We provide detailed experimental analysis of the algorithm and show, in particular, that on practical examples our method is substancially faster than the brute-force approach consisting in reducing separately r matrices with coefficients in $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$. It is important to note that the method does not pretend to scale to very large r, as the arithmetic complexity of operations in $\mathbb{Z}_{q_1 \cdots q_r}$ becomes problematic. Experiments show, however, that for very large r (up to 100000) our approach is still substancially faster than brute-force.

Computing persistent homology with different coefficients has been mentioned in the literature [12] in order to verify if a persisting feature was due to an actual "hole" (or high-dimensional equivalent) or to torsion (and consequently existed only for a certain coefficient field). However, to the best of our knowledge, this is the first work formalizing the inference of torsion coefficients in the framework of persistent homology and describing an efficient algorithm to compute persistence with various coefficient fields.

2 Multi-Field Persistent Homology

For simplicity, we focus in the following on simplicial homology. However, the approach applies to any type of boundary matrix (defined below).

Background on Simplicial Homology and Persistence: A *simplicial complex* $\mathbf{K}$ on a set of *vertices* $V = \{1, \cdots, n\}$ is a collection of simplices $\{\sigma\}$, $\sigma \subseteq V$, such that $\tau \subseteq \sigma \in \mathbf{K} \Rightarrow \tau \in \mathbf{K}$. The dimension $d = |\sigma| - 1$ of σ is its number of elements minus 1. For a ring $\mathcal{R}$, the group of d-chains, denoted $\mathbf{C}_d(\mathbf{K}, \mathcal{R})$, of $\mathbf{K}$ is the group of formal sums of d-simplices with $\mathcal{R}$ coefficients. The *boundary operator* is a linear operator $\partial_d : \mathbf{C}_d(\mathbf{K}, \mathcal{R}) \to \mathbf{C}_{d-1}(\mathbf{K}, \mathcal{R})$ such that $\partial_d \sigma = \partial_d [v_0, \cdots, v_d] = \sum_{i=0}^{d} (-1)^i [v_0, \cdots, \widehat{v_i}, \cdots, v_d]$, where $\widehat{v_i}$ means v_i is deleted from the list. It will be convenient to consider later the endomorphism $\partial_* : \bigoplus_d \mathbf{C}_d(\mathbf{K}, \mathcal{R}) \to \bigoplus_d \mathbf{C}_d(\mathbf{K}, \mathcal{R})$ extended by linearity to the external sum of chain groups. Denote by $\mathbf{Z}_d(\mathbf{K}, \mathcal{R})$ and $\mathbf{B}_{d-1}(\mathbf{K}, \mathcal{R})$ the kernel and the image of ∂_d respectively. Observing $\partial_d \circ \partial_{d+1} = 0$, we define the d^{th} homology group $\mathbf{H}_d(\mathbf{K}, \mathcal{R})$ of $\mathbf{K}$ by the quotient $\mathbf{H}_d(\mathbf{K}, \mathcal{R}) = \mathbf{Z}_d(\mathbf{K}, \mathcal{R}) / \mathbf{B}_d(\mathbf{K}, \mathcal{R})$.

If $\mathcal{R}$ is the *ring of integers* $\mathbb{Z}$, $\mathbf{H}_d(\mathbf{K},\mathbb{Z})$ is an abelian group and, according to the *fundamental theorem of finitely generated abelian groups* [10], admits a *primary decomposition*: $\mathbf{H}_d(\mathbf{K},\mathbb{Z}) \cong \mathbb{Z}^{\beta_d(\mathbb{Z})} \bigoplus_{q \text{ prime}} \left(\mathbb{Z}_{q^{k_1}} \oplus \cdots \oplus \mathbb{Z}_{q^{k_{t(d,q)}}}\right)$ for uniquely defined integer $\beta_d(\mathbb{Z})$, called the d^{th} *integral Betti number*, and integers $t(d,q) \geq 0$ and $k_i > 0$ for every prime number q. If $t(d,q) > 0$, the integers $q^{k_1}, \cdots, q^{k_{t(d,q)}}$ are called *torsion coefficients*, and they admit q as unique *prime divisor*. Intuitively, in dimension 0, 1 and 2, the integral Betti numbers count the number of connected components, the number of holes and the number of voids respectively. The torsion coefficients represent non-orientable twisting of different order of the shape. If $\mathcal{R}$ is a *field* $\mathbb{F}$, $\mathbf{H}_d(\mathbf{K},\mathbb{F})$ is a vector-space and decomposes into $\mathbf{H}_d(\mathbf{K},\mathbb{F}) \cong \mathbb{F}^{\beta_d(\mathbb{F})}$, where $\beta_d(\mathbb{F})$ is the d^{th} *field Betti number*. The field Betti numbers $(\beta_d(\mathbb{F}))_d$ are entirely determined by the characteristic of $\mathbb{F}$ and the inegral homology (see Section 5); hence, the integral homology is more informative than homology in $\mathbb{F}$.

A *filtration* of a complex is a function $f : \mathbf{K} \to \mathbb{R}$ satisfying $f(\tau) \leq f(\sigma)$ whenever $\tau \subseteq \sigma$. The sequence $[\sigma_i]_{i=1,\cdots,m}$ sorted according to increasing f values induces the sequence of inclusions $\emptyset = \mathbf{K}_0 \subsetneq \mathbf{K}_1 \subsetneq \cdots \subsetneq \mathbf{K}_{m-1} \subsetneq \mathbf{K}_m = \mathbf{K}$, $\mathbf{K}_i = \mathbf{K}_{i-1} \cup \{\sigma_i\}$, and the sequence of d-homology groups $0 = \mathbf{H}_d(\mathbf{K}_0,\mathcal{R}) \to \mathbf{H}_d(\mathbf{K}_1,\mathcal{R}) \to \cdots \to \mathbf{H}_d(\mathbf{K}_{m-1},\mathcal{R}) \to \mathbf{H}_d(\mathbf{K}_m,\mathcal{R}) = \mathbf{H}_d(\mathbf{K},\mathcal{R})$ connected by homomorphisms. When $\mathcal{R}$ is a field, the later sequence admits a decomposition in terms of intervals $\{(i,j)\}$, called an *indexed persistence diagram*, where a pair (i,j) is interpreted as a homology feature that *is born* at index i and *dies* at index j. Computing persistent homology consists in computing the interval decomposition and hence the persistence diagram. We refer to [10] for an introduction to homology and to [5] for an introduction to persistent homology.

We call the algorithmic problem of computing persistent homology with various coefficient fields *multi-field persistent homology*. Computing multi-field persistence allows us to infer a more informative description of the topology of a space, compared to persistence in a single field (see Section 5 for details). For a complex of size m, we know that the persistence diagram for any coefficient field contains at most m pairs. When computing multi-field persistent homology for r coefficient fields, denote by m' the total number of *distinct* pairs in all persistence diagrams. In practice, the fields are $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ for the r first prime numbers $q_1, \cdots, q_r$, where q_r is an upper bound on the prime divisors of the torsion coefficients of the integral homology of the space, which are usually small (see Section 5). The quantity m' satisfies $m' \leq r \times m$ but in practice we observe that $m' \approx m$. We design in the following an algorithm for the multi-field persistence problem whose complexity depends mostly on m'. It is however an interesting open problem to exhibit a "natural" example where m' is must larger than m and/or the prime divisors of the torsion coefficients are big (for a fix m).

3 Algorithm for Multi-Field Persistent Homology

For clarity, we focus in this section on the persistent homology algorithm as presented in [5], which consists in a reduction to column echelon form (defined

later) of a matrix. All other persistent homology algorithms are based on similar reductions, and our approach adapts directly to them. In the following, $\mathbb{Z}_n$ denotes the ring $(\mathbb{Z}_n, +, \times)$ for any integer $n \geq 1$. and $\mathbb{Z}_n^\times$ the subset of invertible elements for $\times$. If exists, we denote the inverse of $x \in \mathbb{Z}_n$ by x^{-1}.

In computer algebra, working modulo small prime numbers is usually desireable in order to reduce coefficient growth. Our work goes the otherway around: we introduce tools to reduce a family of r matrices with coefficients in the fields $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ respectively, by means of a single reduction of a matrix with coefficients in $\mathbb{Z}_{q_1 \cdots q_r}$. We give theoretical and experimental evidence that, for reasonable values of r, our algorithm is significantly more efficient than the brute-force approach consisting in reducing the r matrices separately.

Persistent Homology Algorithm: For an $m \times m$ matrix $\mathbf{M}$, denote by C_j the j^{th} column of $\mathbf{M}$, $1 \leq j \leq m$, and $C_j[k]$ its k^{th} coefficient. Let $\text{low}(j)$ denote the row index of the lowest non-zero coefficient of C_j. If the column j is entirely zero, $\text{low}(j)$ is undefined. We say that $\mathbf{M}$ is in *reduced column echelon form* if $\text{low}(j) \neq \text{low}(j')$ for every non-zero columns C_j and $C_{j'}$ with $j \neq j'$.

Let $\mathbf{K} = [\sigma_i]_{i=1\cdots m}$ be a filtered complex. Its boundary matrix $\mathbf{M}_\partial$ is the $m \times m$ matrix, with $\mathbb{F}$ coefficients, of the endomorphism ∂_* in the basis $\{\sigma_1, \cdots, \sigma_m\}$ of $\bigoplus_d \mathbf{C}_d(\mathbf{K}, \mathbb{F})$. The basis is ordered according to the filtration. It is a matrix with $\{-1, 0, 1\}$ coefficients, where 0 and 1 are the identities for $+$ and $\times$ in $\mathbb{F}$ respectively, and -1 is the inverse of 1 in $\mathbb{F}$. The persistent homology algorithm consists in a left-to-right reduction to column echelon form of $\mathbf{M}_\partial$: we denote by $\mathbf{R}$ the matrix we reduce, with columns C_j, which is initially equal to $\mathbf{M}_\partial$. The algorithm returns the *(indexed) persistence diagram*, which is the set of pairs $\{(\text{low}(j), j)\}$ in the reduced column echelon form of the matrix.

Data: Boundary matrix $\mathbf{R} \leftarrow \mathbf{M}_\partial$, persistence diagram $\mathcal{P} \leftarrow \emptyset$
Output: Persistence diagram $\mathcal{P} = \{(i, j)\}$

```
for j = 1, ··· , m do
    while there exists j' < j with low(j') = low(j) do
        k ← low(j);
        C_j ← C_j − (C_j[k] × C_j'[k]^{-1}) · C_j';
    end
    if C_j ≠ 0 then P ← P ∪ {(low(j), j)}
end
```

The reduced form of the matrix is not unique, but the pairs (i, j) such that $i = \text{low}(j)$ in the column echelon form are [5]. The algorithm requires $O(m^3)$ arithmetic operations in $\mathbb{F}$.

3.1 Modular Reconstruction for Elementary Matrix Operations:

We present a particular case of the *Chinese Remainder Theorem* [7]: *For a family $\{q_1, \cdots, q_r\}$ of r distinct prime numbers, there exists a ring isomorphism $\psi : \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r} \to \mathbb{Z}_{q_1 \cdots q_r}$. The isomorphisms ψ and ψ^{-1} can be computed in $O(r)$ arithmetic operations in $\mathbb{Z}_{q_1 \cdots q_r}$.*

Let $[r]$ refer to the set $\{1, \cdots, r\}$. For a family of r distinct prime numbers $\{q_1, \cdots, q_r\}$, and a subset of indices $S \subseteq [r]$, Q_S refers to $\prod_{s\in S} q_s$, and we write simply $Q = Q_{[r]}$. We define the function $\psi_S : \prod_{s\in S} \mathbb{Z}_{q_s} \to \mathbb{Z}_{Q_S}$ realizing the isomorphism of the Chinese Remainder Theorem for the subset $\{q_s\}_{s\in S}$ of primes, and we write simply ψ for $\psi_{[r]}$. For a family of elements $u_s \in \mathbb{Z}_{q_s}, s \in S$, we denote the corresponding $|S|$-uplet $(u_s)_{s\in S} \in \prod_{s\in S} \mathbb{Z}_{q_s}$.

Finally, we recall *Bezout's lemma* [7]: *For two integers a and b, not both 0, there exist integers v and w such that $va + wb = \gcd(a, b)$, the greatest common divisor of a and b, with $|v| < |b/\gcd(a, b)|$ and $|w| < |a/\gcd(a, b)|$. The Bezout's coefficients (v, w) can be computed with the extended Euclidean algorithm [7].*

Elementary Column Operations: We are given a family of distinct prime numbers $\{q_1, \cdots, q_r\}$, and their product $Q = q_1 \cdots q_r$. Let $\mathbf{M}_Q$ be a matrix with coefficients in the ring $\mathbb{Z}_Q$. Denoting $\psi^{-1} : \mathbb{Z}_Q \to \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ the isomorphism of the Chinese Remainder Theorem, and $\pi_s : \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r} \to \mathbb{Z}_{q_s}$ the projection on the s^{th} coordinate, we call *projection of $\mathbf{M}_Q$ onto $\mathbb{Z}_{q_s}$*, denoted $\mathbf{M}_Q(\mathbb{Z}_{q_s})$, the matrix $\mathbf{M}_{q_s}$ with $\mathbb{Z}_{q_s}$ coefficients, obtained by applying $\pi_s \circ \psi^{-1}$ to each coefficient of $\mathbf{M}_Q$. Conversely, given r $(m \times m)$-matrices $\mathbf{M}_{q_1}, \cdots, \mathbf{M}_{q_r}$ with coefficients in $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ respectively, there exists a unique matrix $\mathbf{M}_Q$ with $\mathbb{Z}_Q$ coefficients such that for every s the projection of $\mathbf{M}_Q$ onto $\mathbb{Z}_{q_s}$ is $\mathbf{M}_{q_s}$. This is simply a matrix version of the Chinese remainder theorem. *Elementary column operations* on $\mathbf{M}_q$ with $\mathbb{Z}_q$ coefficients are of three kinds:

(i) exchange Col k and Col ℓ
(ii) multiply Col k by $-1 \in \mathbb{Z}_q$
(iii) replace Col k by (Col k)+ $x\times$(Col ℓ), for $x \in \mathbb{Z}_q$.

For an elementary column operation $(*)$ (i.e. an operation of type (i), (ii) or (iii) applied to columns k (and ℓ)), we denote by $(*) \circ \mathbf{M}_q$ the result of applying $(*)$ to $\mathbf{M}_q$. In this section, we introduce algorithms to run elementary column operations simultaneously on the matrices $(\mathbf{M}_{q_s})_{s=1,\cdots,r}$ by performing "partial column operations" on $\mathbf{M}_Q$. Specifically, for an elementary column operation $(*)$ and a subset of indices $S \subseteq [r]$, we call *partial column operation* on $\mathbf{M}_Q$ the operation transforming $\mathbf{M}_Q$ into $\mathbf{M}'_Q$ such that: for every $s \notin S$, the projection onto $\mathbb{Z}_{q_s}$ satisfies $\mathbf{M}_Q(\mathbb{Z}_{q_s}) = \mathbf{M}'_Q(\mathbb{Z}_{q_s}) = \mathbf{M}_{q_s}$ and for every $s \in S$, the projection onto $\mathbb{Z}_{q_s}$ satisfies $\mathbf{M}'_Q(\mathbb{Z}_{q_s}) = (*) \circ \mathbf{M}_{q_s}$.

As the correspondence $\psi : \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r} \to \mathbb{Z}_Q$ is a ring homomorphism, it satisfies the properties: $\psi(u_1, \cdots, u_r)+\psi(v_1, \cdots, v_r)\times\psi(w_1, \cdots, w_r) = \psi(u_1+v_1\times w_1, \cdots, u_r+v_r\times w_r)$ and we can compute addition and multiplication componentwise in $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ using addition and multiplication in $\mathbb{Z}_Q$. In order to compute partial column operations, we first introduce the set of *partial identities*, which are coefficients that allow us to proceed to the partial column operations of type (i) and (ii). Secondly, as the rings $\mathbb{Z}_{q_s}$ are fields, we need to compute the multiplicative inverse of an element, that is used as multiplicative coefficient x in elementary column operation (iii). As $\mathbb{Z}_Q$ is not a field, inversion is not possible, and we introduce the concept of *partial inverse* to overcome this difficulty. In the following, the term "arithmetic operation" refers to any op-

eration $\{+, -, \times, \gcd(\cdot,\cdot), \cdot \mod Q_S, \text{Extended Euclidean algorithm}\}$ on integer smaller than Q. Note they do not have constant time complexity for large Q.

Partial Identity and Partial Inverse: Given a subset of indices $S \subseteq [r]$, we define the *partial identities w.r.t. S*, denoted L_S, by $L_S = \psi(\delta_{1,S}, \cdots, \delta_{r,S})$ where the symbol $\delta_{t,S} \in \mathbb{Z}_{q_t}$ is equal to 1 if $t \in S$ and to 0 otherwise. For any $S \subseteq [r]$, the partial identity L_S can be constructed in $O(r)$ arithmetic operations in $\mathbb{Z}_Q$ by evaluating ψ on $(\delta_{1,S}, \cdots, \delta_{r,S})$. However, it is important to notice that if $S = [r]$, $L_{[r]} = \psi(1, \cdots, 1) = 1$, because ψ is a ring isomorphism, and L_r is computed in time $O(1)$.

Knowing the partial identities, we can implement the partial column operations (i) and (ii) for a set of indices S. Partial column operation (i) is implemented by replacing column k by $(\text{Col } k \times L_{[r]\setminus S} + \text{ Col } \ell \times L_S)$ and column ℓ by $(\text{Col } \ell \times L_{[r]\setminus S} + \text{ Col } k \times L_S)$. Partial column operation (ii) is implemented by multiplying column k by $L_{[r]} - 2 \times L_S$.

We define now the *partial inverse* of an element in the ring $\mathbb{Z}_Q$:

Definition 1 (Partial Inverse) *Given a set $S \subseteq [r]$ of indices, the* partial inverse *of $x = \psi(u_1, \cdots, u_r)$ with regard to S is the element $\overline{x}^S \in \mathbb{Z}_Q$:*

$$\overline{x}^S = \psi(\overline{u_1}^S, \cdots, \overline{u_r}^S), \quad \textit{with } \overline{u_s}^S = \begin{cases} u_s^{-1} & \textit{if } s \in S \textit{ and } u_s \in \mathbb{Z}_{q_s}^\times \\ 0 & \textit{otherwise} \end{cases}$$

Using elementary algebra (see [2] for details) we prove:

Proposition 2 (Partial Inverse Construction) *For $x = \psi(u_1, \cdots, u_r) \in \mathbb{Z}_Q$ and $S \subseteq [r]$,*

(1) $\gcd(x, Q_S) = Q_R$ for some $R \subseteq S$ and for all $s \in S$, u_s is invertible in $\mathbb{Z}_{q_s}$ iff $s \notin R$; we denote $T = S \setminus R$.
(2) The Bezout's identity for x and Q_T gives $vx + wQ_T = 1$, where v satisfies $v \mod Q_T = \psi_T((u_s^{-1})_{s\in T})$
(3) $\overline{x}^S = \left[\psi_T((u_s^{-1})_{s\in T}) \times L_T \mod Q\right] \in \mathbb{Z}_Q$, where L_T is the partial identity w.r.t T.

We deduce directly an algorithm to compute the partial inverse of x w.r.t S if Q_S is given: compute $Q_R = \gcd(x, Q_S)$ and $Q_T = Q_S/Q_R$, then v using the extended Euclidean algorithm and finally $\overline{x}^S = (v \mod Q_T) \times L_T \mod Q$. Computing the partial identity L_T requires $O(r)$ arithmetic operations in $\mathbb{Z}_Q$, but is constant if $T = [r]$, which happens iff $S = [r]$ and x is invertible in $\mathbb{Z}_Q$. Consequently, computing $\overline{x}^S$ requires $O(r)$ arithmetic operations in general, but only $O(1)$ arithmetic operations in the later case.

3.2 Modular Reconstruction for Multi-Field Persistent Homology

Let $\mathbf{K}$ be a filtered complex with m simplices. Define $\mathbf{M}_\partial(\mathbb{Z}_{q_s})$ to be the $(m \times m)$ boundary matrix of $\mathbf{K}$ with $\mathbb{Z}_{q_s}$ coefficients. Define $\mathbf{M}$ to be the $(m \times m)$ matrix with $\mathbb{Z}_Q$ coefficients such that the projection of $\mathbf{M}$ onto $\mathbb{Z}_{q_s}$ is equal to $\mathbf{M}_\partial(\mathbb{Z}_{q_s})$,

for all $s \in [r]$. Note that the matrices $\mathbf{M}$ and $\mathbf{M}_\partial(\mathbb{Z}_{q_s})$, for any s, are "identical" matrices in the sense that they contain 0, 1 and -1 coefficients at the same positions, where 0, 1 and -1 refer respectively to elements of $\mathbb{Z}_Q$ and $\mathbb{Z}_{q_s}$.

We reduce a matrix $\mathbf{R}$ which is initially equal to $\mathbf{M}$. Denote by C_j the j^{th} column of $\mathbf{R}$. Define $\text{low}(j, Q_S)$ to be the index of the lowest element of C_j such that $C_j[\text{low}(j, Q_S)] \mod Q_S \neq 0$. In particular, $\text{low}(j, q_s)$ is equal to the index of the lowest non-zero element of column j in the projection $\mathbf{R}(\mathbb{Z}_{q_s})$. After iteration j, we say that the columns $C_1, \cdots, C_j$ are *reduced*. We maintain, for every reduced column C_j, the collection of "lowest indices" i as a set $\mathcal{L}(j) = \{(i, Q_S)\}$ satisfying:

- For every $(i, Q_S) \in \mathcal{L}(j)$, $i = \text{low}(j, Q_S)$
- For every $(i, Q_S), (i', Q_{S'}) \in \mathcal{L}(j)$, either $i = i'$ and $S = S'$, or $i \neq i'$ and $S \cap S' = \emptyset$
- $\cup_{(i,Q_S)\in\mathcal{L}(j)} S = [r]$

The algorithm returns the set of triplets $\mathcal{P} = \{(i, j, Q_S)\}$ such that $i = \text{low}(j)$ in the column echelon form of the matrix $\mathbf{M}_\partial(\mathbb{Z}_{q_s})$ iff $s \in S$, or, equivalently, $(i, Q_S) \in \mathcal{L}(j)$ once C_j has been reduced. This is a compact encoding of the persistence diagrams of the filtered complex in persistent homology with all coefficient fields. We call it *multi-field persistence diagram*.

Data: Matrix $\mathbf{R} = \mathbf{M}$
Output: Multi-field persistence diagram $\mathcal{P} = \{(i, j, Q_S)\}$

1 **for** $j = 1, \cdots, m$ **do**
2 $\quad Q_S \leftarrow Q_{[r]}$;
3 $\quad$ **while** *low*(j, Q_S) *is defined* **do**
4 $\quad\quad k \leftarrow \text{low}(j, Q_S)$; $\quad Q_T \leftarrow Q_S / \gcd(C_j[k], Q_S)$;
5 $\quad\quad$ **while** *there exists* $j' < j$ *with* $(i, Q_{T'}) \in \mathcal{L}(j')$ *satisfying*
6 $\quad\quad\quad$ $[i = low(j, Q_S)$ *and* $\gcd(Q_{T'}, Q_T) > 1]$ **do**
7 $\quad\quad\quad\quad C_j \leftarrow C_j - \left(C_j[k] \times \overline{C_{j'}[k]}^T\right) \cdot C_{j'}$;
8 $\quad\quad\quad\quad Q_T \leftarrow Q_T / \gcd(Q_{T'}, Q_T)$;
9 $\quad\quad$ **end**
10 $\quad\quad$ **if** $Q_T \neq 1$ **then** $\mathcal{P} \leftarrow \mathcal{P} \cup \{(k, j, Q_T)\}$; $\quad Q_S \leftarrow Q_S / Q_T$;
11 $\quad$ **end**
12 **end**

The $\{\mathcal{L}(j)\}_j$ form an index table that we maintain implicitely. At iteration j of the `for` loop, we use Q_S for the product of all prime numbers $\prod_{s\in S} q_s$ for which the column j in $\mathbf{R}(\mathbb{Z}_{q_s})$ has not yet been reduced.

Correctness: First, note that all operations processed on $\mathbf{R}$ correspond to left-to-right elementary column operations in the matrices $\mathbf{R}(\mathbb{Z}_{q_s})$ for all $s \in [r]$. By definition of the partial inverse, the column operation in line 7 can only reduce the value of $\text{low}(j, Q_S)$. Moreover, one iteration of the `while` loop in line 3 either strictly reduces Q_S by dividing it by Q_T (in line 10) or set $(C_j[k] \mod Q_S)$ to zero, hence reducing strictly $\text{low}(j, Q_S)$. The later case happens when Q_T is set to 1 in line 8. Consequently, the algorithm terminates.

We prove recursively, on the numbers of columns, that each of the matrix $\mathbf{R}(\mathbb{Z}_{q_s})$ gets reduced to column echelon form. We fix an arbitrary field $\mathbb{Z}_{q_s}$: suppose that the $j-1$ first columns of $\mathbf{R}(\mathbb{Z}_{q_s})$ have been reduced at the end of iteration $j-1$ of the `for` loop in line 1. We prove that at the end of the j^{th} iteration of the `for` loop in line 1, the j first columns of the matrix $\mathbf{R}(\mathbb{Z}_{q_s})$ are reduced. Consider two cases. First suppose there is a triplet $(i, j, Q_T) \in \mathcal{P}$ for some $i < j$ and Q_T satisfying $q_s \mid Q_T$. This implies that the algorithm exits the `while` loop in line 5 with $q_s \mid Q_S$ (because by definition of Q_T, in line 4, $Q_T \mid Q_S$) and there is no $j' < j$ such that $[\text{low}(j', Q_{T'}) = \text{low}(j, Q_S)$ and $\gcd(Q_{T'}, Q_T) > 1]$. This in particular implies that there is no $j' < j$ such that $\text{low}(j', q_s) = \text{low}(j, q_s)$ and column j is reduced in $\mathbf{R}(\mathbb{Z}_{q_s})$.

Secondly, suppose that there is no such pair (i, j, Q_T) in $\mathcal{P}$, with q_s dividing Q_T. Consequently, during all the computation of the `while` loop in line 3, $q_s \mid Q_S$. When exiting this `while` loop, $\text{low}(j, Q_S)$ is undefined, implying in particular that $\text{low}(j, q_s)$ is undefined and column j of $\mathbf{R}(\mathbb{Z}_{q_s})$ is zero, and hence reduced.

4 Output-Sensitive Complexity Analysis

Arithmetic Complexity Model for Large Integers: During the reduction algorithm we perform arithmetic operations on big integers, for which we describe a complexity model [7]. Suppose that on our architecture, a memory word is encoded on w bits (on modern architectures, w is usually 64). Computer chips contains Arithmetic Logic Units that allow arithmetic operations on a 1-memory word integer in $O(1)$ machine cycles. Let the *length* of an integer z be defined by: $\lambda(z) = \lfloor \log_2 z/\mathsf{w} \rfloor + 1$, i.e. by the number of memory words necessary to encode z. We express the arithmetic complexity as a function of the length. For any positive integer z of length $\lambda(z) = \mathsf{B}$, operations in $\mathbb{Z}_z$ cost $\mathsf{A}_+(z) = O(\mathsf{B})$ for addition, $\mathsf{A}_\times(z) = O(M(\mathsf{B}))$ for multiplication and $\mathsf{A}_\div(z) = O(M(\mathsf{B}) \log \mathsf{B})$ for (extended) Euclidean algorithm, inversion and division, where $M(n)$ is a monotonic upper bound on the number of word operations necessary to multiply two integers of length B. By a result of [6], $M(\mathsf{B}) = O(\mathsf{B} \log \mathsf{B}\ 2^{O(\log^* \mathsf{B})})$, where $\log^* n$ is the iterated logarithm of n. In the following, we write $\mathsf{A}(z)$ for a bound on the complexity of arithmetic operations on integers smaller than z.

In the case of multi-field persistent homology, we are interested in the value of λ for an element in $\mathbb{Z}_Q$, $Q = q_1 \cdots q_r$, in the case where $\{q_1, \cdots, q_r\}$ are the first r prime numbers. We know [11] that $\ln Q < 1.01624 q_r$ and $q_r < r \ln(r \ln r)$ for $r \geq 6$. Consequently, $\lambda(Q) < \lfloor 1.46613 r \ln(r \ln r)/\mathsf{w} \rfloor + 1$. Note that $\lambda(Q) \ll r$ for $r \ln r \ll e^{\mathsf{w}}$, which is a reasonable assumption.

Complexity of the Modular Reconstruction Algorithm: Let $\mathbf{K}$ be a filtered complex of size m. The persistent homology algorithm described in Section 3, applied on $\mathbf{K}$ with coefficients in a field $\mathbb{F}$, requires $O(m^3)$ operations in $\mathbb{F}$. For a field $\mathbb{Z}_q$ these operations take constant time and the algorithm has complexity $O(m^3)$. The output of the algorithm is the persistence diagram, which has size $O(m)$ for any field.

For a set of prime numbers $\{q_1, \cdots, q_r\}$, let m' be the total number of distinct pairs in all persistence diagrams for the persistent homology of $\mathbf{K}$ with coefficient fields $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$. We express the complexity of the modular reconstruction algorithm in terms of the size of its output (i.e. the multi-field persistence diagram of size m'), the number of fields r and the arithmetic complexity $\mathsf{A}(Q)$. First, note that, for a column j' in the reduced form of $\mathbf{R}$, the size of $\mathcal{L}(j')$ is equal to the number of triplets of the multi-field persistence diagram with death index j'; denote this quantity by $|\mathcal{L}(j')|$. Hence, when reducing column $j > j'$, the column $C_{j'}$ is involved in a column operation $C_j \leftarrow C_j + \alpha \cdot C_{j'}$ at most $|\mathcal{L}(j')|$ times. Consequently, reducing C_j requires $O(\sum_{j'<j} |\mathcal{L}(j')|) = O(m')$ column operations. There is a total number of $O(m \times m')$ column operations to reduce the matrix, each of them being computed in time $O(m \times \mathsf{A}(Q))$.

Computing the partial inverse of an element $x \in \mathbb{Z}_Q$ takes time $O(r \times \mathsf{A}(Q))$ in the general case, and only $O(\mathsf{A}(Q))$ if x is invertible in $\mathbb{Z}_Q$. The partial inverse of an element $x = C_j[k]$ is computed only if there is a pair $(k, Q_T) \in \mathcal{L}(j)$. This element is not invertible in $\mathbb{Z}_Q$ iff $|\mathcal{L}(j)| > 1$. There are consequently $O(|m' - m|)$ non-invertible elements x that are at index $\mathrm{low}(j, Q_T)$ in some column j, for some Q_T. If we store the partial inverses when we compute them, the total complexity for computing all partial inverses in the modular reconstruction algorithm is $O((m + r \times (m' - m) \times \mathsf{A}(Q))$. We conclude that the total cost of the modular reconstruction algorithm for multi-field persistent homology is $O(\left[r \times (m' - m) + m^2 m'\right] \times \mathsf{A}(Q)) = O(\left[r \times (m' - m) + (m')^3\right] \times \mathsf{A}(Q))$, while the brute-force algorithm, consisting in computing persistence separately for every field $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$ requires $O(r \times m^3)$ operations.

Note that asymptotically in r, one arithmetic operation in $\mathbb{Z}_{Q_{[r]}}$ becomes more costly than r distinct arithmetic operations in $\mathbb{Z}_{q_1}, \cdots, \mathbb{Z}_{q_r}$, in which case the modular reconstruction approach developed in this article becomes worse than brute-force (even when m' and m are close). This however happens for extremely big values of r (see Section 5) and has no incidence on practical cases.

Remark: On all datasets considered in our experiments, we have found no example where m' was significantly bigger than m. However, it is unclear whether many "short-lived torsion" might appear in general. We prove in a long version of the paper [2] that this is not an issue, by giving a finer complexity analysis of the algorithm in terms of index persistence $|j - i|$ of the pairs (i, j) in the persistence diagram.

5 Experiments

In this section, we report on the performance of the modular reconstruction algorithm for multi-field persistent homology. Our implementation is in `C++`, and we use the `GMP` library for storing large integers. All timings are measured on a 64 bits Linux machine with 3.00 GHz processor and 32 GB RAM. All timings are averaged over 10 independent runs. We compute the persistent homology of Rips complexes [5] built on a variety of both real and synthetic datasets. We use the *compressed annotation matrix* implementation of persistence [1] for its

Data	$\lvert\mathcal{P}\rvert$	D	d	r	$\lvert\mathcal{K}\rvert$	T_1	R_1	T_{50}	R_{50}	T_{100}	R_{100}	T_{200}	R_{200}
Bud	49,990	3	2	0.09	$127 \cdot 10^6$	96.3	0.51	110.3	22.2	115.9	42.3	130.7	75.0
Bro	15,000	25	?	0.04	$142 \cdot 10^6$	123.8	0.41	143.5	17.8	150.2	34.0	174.5	58.5
Cy8	6,040	24	2	0.8	$193 \cdot 10^6$	121.2	0.63	134.6	28.2	139.2	54.6	148.8	102.2
Kl	90,000	5	2	0.25	$114 \cdot 10^6$	78.6	0.52	89.3	23.0	93.0	44.1	105.2	78.0
S3	50,000	4	3	0.65	$134 \cdot 10^6$	125.9	0.40	145.7	17.2	152.6	32.8	177.6	50.3

Fig. 1. Timings of the modular reconstruction algorithm vs brute-force.

efficiency and stability. **Bud** is a set of points sampled from the surface of the *Stanford Buddha* in $\mathbb{R}^3$. **Bro** is a set of 5×5 *high-contrast patches* derived from natural images, interpreted as vectors in $\mathbb{R}^{25}$, from the Brown database (with parameter $k = 300$ and cut 30%) [3]. **Cy8** is a set of points in $\mathbb{R}^{24}$, sampled from the space of conformations of the cyclo-octane molecule [9], which is the union of two intersecting surfaces. **Kl** is a set of points sampled from the surface of the figure eight Klein Bottle embedded in $\mathbb{R}^5$. Finally **S3** is a set of points distributed on the unit 3-sphere in $\mathbb{R}^4$. Datasets are listed in Figure 1 with the size of points sets $|\mathcal{P}|$, the ambient dimension D and intrinsinc dimension d of the sample points (if known), the parameter r for the Rips complex and the size of the complex $|\mathcal{K}|$. The values T_r for $r \in \{1, 50, 100, 200\}$ refers to the running time of the modular reconstruction algorithm for the r first prime numbers, and R_r refers to the ratio between the brute-force approach and the modular reconstruction algorithm.

Interpretation of the Results: Surprisingly, we have observed that, on all experiments, the number of differences between persistence diagrams with various coefficient fields was extremely small. As a consequence, $m' - m$ can be considered as a very small constant in our experiments (≤ 10). We have also observed that these differences appeared for small prime numbers q_s.

Figure 1 presents the timings of the modular reconstruction approach for a variety of simplicial complexes ranging between 114 and 193 million simplices. We note that from $r = 1$ to $r = 200$ prime numbers, the time for computing multi-field persistence using the modular reconstruction approach only increases by 23 to 41%, when the brute-force approach requires about 200 times more time. This difference appears in the speedup expressed by the ratio R_r. For $r = 1$, the modular reconstruction approach is about twice slower than the standard persistent homology algorithm in one field, because modular reconstruction is a more complex procedure and deals, in our implementation, with `GMP` integers that are slower than the classic `int` used in the standard persistent homology algorithm. However, this difference fades away as soon as $r > 1$ and the modular reconstruction is significantly more efficient than brute-force: it is, in particular, between 50.3 and 102.2 times faster for $r = 200$.

Figures 2 and 3 present the evolution of the running time of the modular reconstruction approach and the brute-force approach for an increasing number of fields r (using the first r prime numbers). Persistence is computed for a Rips complex built on a set of 10000 points sampling a Klein bottle, which

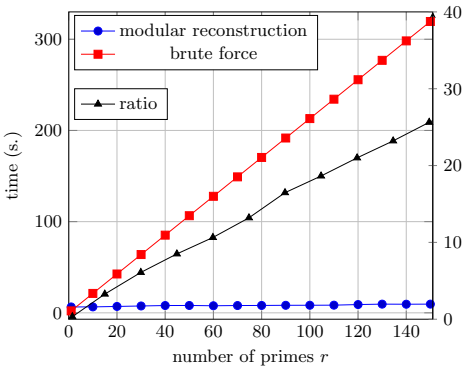

Fig. 2. Timings for the modular reconstruction algorithm and brute force.

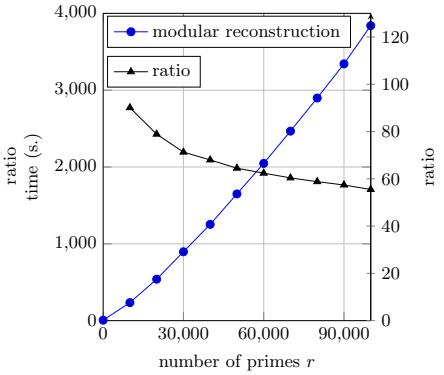

Fig. 3. Asymptotic behavior of modular reconstruction and brute force.

contains torsion in its integral homology, resulting in a simplicial complex of 6.14 million simplices. We analyze the result in terms of the complexity analysis of Section 4. The quantity m is fixed and m' is fixed for $r \geq 2$. The complexity of the brute-force algorithm is $O(r \times m^3)$ and we indeed observe a linear behavior when r increases. The complexity of the modular reconstruction approach is $O(\left[r \times (m' - m) + m'^3\right] \mathsf{A}(Q_{[r]}))$. The part $r \times (m' - m)$ of the complexity is negligeable because $m' - m$ is extremely small. For medium values of r (≤ 150), like in Figure 2, the arithmetic complexity $O(\mathsf{A}(Q_{[r]}))$ increases very slowly because $\lambda(Q_{[r]}) = \left\lfloor \log_2 Q_{[r]}/\mathsf{w} \right\rfloor + 1$ increases slowly. We consequently observe a very slow increasing of the time complexity compare to the one of brute-force.

Figure 3 describes the asymptotic behavior of the modular approach, where the arithmetic operations become costly. We observe that the timings for the modular reconstruction approach follow a convex curve. The convexity comes from the growth of $\lambda(Q_{[r]})$, which is asymptotically $\Theta(r \log r))$ [11]. However, the increasing of the slope is very slow: all along this experiment, we have been unable to reach a value of r for which the modular approach is worse than the brute-force approach. For readability, the timings for the brute-force approach are implicetly represented through their ratio with the modular approach: all along the experiment, for $10000 \leq r \leq 100000$, the modular approach is between 55 and 90 times faster. Based on a linear interpolation of the timings for the brute-force approach, and a polynomial interpolation of the modular reconstruction timings, we expect the modular reconstruction to become worse than brute-force for a number of primes r bigger than 4.9 million. In the case of multi-field persistent homology however, there is no need to take r bigger than 200, because r is related to torsion coefficients (see Section 5), which are small in practice.

Back to Topology: Inference of Torsion For a topological space $\mathbb{X}$, the *Universal Coefficient Theorem for Homology* [10] establishes the relationship between the homology groups $\mathbf{H}_d(\mathbb{X}, \mathbb{Z})$ with $\mathbb{Z}$ coefficients and the homology

groups $\mathbf{H}_d(\mathbb{X}, \mathbb{Z}_q)$ with coefficients in the field $\mathbb{Z}_q$ (of characteristic q), for q prime. We use the following corollary:

Corollary 3 (Universal Coefficient Theorem [10]) *For $\beta_d(\mathbb{Z})$ and $\beta_d(\mathbb{Z}_q)$ the Betti numbers of $\mathbf{H}_d(\mathbb{X}, \mathbb{Z})$ and $\mathbf{H}_d(\mathbb{X}, \mathbb{Z}_q)$ respectively, and $t(j, q)$ the number of $\mathbb{Z}_{q^{k_i}}$ summands in the primary decomposition of $\mathbf{H}_j(\mathbb{X}, \mathbb{Z})$, we have:*

$$\beta_d(\mathbb{Z}_q) = \beta_d(\mathbb{Z}) + t(d, q) + t(d-1, q)$$

Suppose $\{q_1, \cdots, q_r\}$ are the first r prime numbers and q_r is a strict upper bound on the prime divisors of the torsion coefficients of $\mathbb{X}$. Consequently, according to Corollary 3, $\beta_d(\mathbb{Z}_{q_r}) = \beta_d(\mathbb{Z})$ for all dimensions d. Moreover, we know [10] that there is no torsion in 0-homology (i.e. $t(0, q) = 0$ for all primes q). Given the Betti numbers of $\mathbb{X}$ in all fields $\mathbb{Z}_{q_s}, 1 \leq s \leq r$, we deduce from Corollary 3 the recurrence formula $t(d, q_s) = \beta_d(\mathbb{Z}_{q_s}) - \beta_d(\mathbb{Z}_{q_r}) - t(d-1, q_s)$, from which we compute the value of $t(d, q)$ for every dimension d and prime q. For any dimension d, we consequently infer the integral Betti numbers and the number $t(d, q)$ of $\mathbb{Z}_{q^{k_i}}$ summands in the primary decomposition of $\mathbf{H}_d(\mathbb{X}, \mathbb{Z})$. We note however that the powers k_i from the decomposition remain unknown.

We describe in a long version of this article [2] a representation of multi-field persistence diagrams with torsion coefficients. We also describe an efficient algorithm to compute distances between them.

References

1. Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In *ESA*, pages 695–706, 2013.
2. Jean-Daniel Boissonnat and Clément Maria. Computing persistent homology with various coefficient fields in a single pass. RR-8436, INRIA, December 2013.
3. Gunnar Carlsson, Tigran Ishkhanov, Vin Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, pages 1–12, 2008.
4. David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
5. Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.
6. Martin Fürer. Faster integer multiplication. *SIAM J. Comput.*, 2009.
7. Joachim Von Zur Gathen and Jurgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
8. Clément Maria. GUDHI, simplex tree and persistent cohomology packages. `https://project.inria.fr/gudhi/software/`.
9. S. Martin, A. Thompson, E.A. Coutsias, and J. Watson. Topology of cyclo-octane energy landscape. *J Chem Phys*, 132(23):234115, 2010.
10. James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984.
11. J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. 6:64–94, 1962.
12. Afra Zomorodian and Gunnar E. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.